Working with Universal Dependencies

Martin Popel
popel@ufal.mff.cuni.cz

ÚFAL (Institute of Formal and Applied Linguistics)
Charles University, 2017-03-13

"I SPEND A LOT OF TIME ON THIS TASK. I SHOULD WRITE A PROGRAM AUTOMATING IT!"

THEORY:

WORK

WRITING CODE

WORK ON ORIGINAL TASK

AUTOMATION TAKES OVER

FREE TIME

TIME

REALITY:

WORK

WRITING CODE

DEBUGGING

ONGOING DEVELOPMENT

RETHINKING

NO TIME FOR ORIGINAL TASK ANYMORE

TIME

https://xkcd.com/1319/
Overview

- Universal Dependencies
- A thought on simplicity
- CoNLL 2017 Shared Task
- Tools for UD
- Udapi
Universal Dependencies

- cross-linguistically consistent treebank annotation
- de facto standard for dependency annotation
- builds on:
  Stanford Dependencies, CoNLL, Google UPOS + UDT, HamleDT, Interset
- balances:
  details vs. simplicity (linguists vs. NLP applications)
  adequacy for a given lang. vs. cross-ling
- started in 2014 (kick-off, guidelines v1)
- first release in 2015, new release each 6 months
UDv2.0 released on 2017-03-01:
145 contributors, 70 treebanks, 50 langs, 12M words

457 papers about Universal Dependencies, 582 citations

(source: Google Scholar, March 2017)
Universal Dependencies v2

Executive summary of changes from v1 to v2

- Tokenization and word segmentation
- Morphology
  - General principles
  - Universal POS tags (single document)
  - Universal features (single document)
  - Language-specific features
  - Conversion from other tagsets
- Syntax
  - General principles
  - Basic dependencies
    - Simple clauses
    - Nominals
    - Complex clauses
    - Other constructions
  - Enhanced dependencies
  - Universal dependency relations (single document)
  - Language-specific relations
- CONLL-U format
# Universal POS tags

<table>
<thead>
<tr>
<th>Open class words</th>
<th>Closed class words</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ</td>
<td>ADP</td>
<td>PUNCT</td>
</tr>
<tr>
<td>ADV</td>
<td>AUX</td>
<td>SYM</td>
</tr>
<tr>
<td>INTJ</td>
<td>CCONJ</td>
<td></td>
</tr>
<tr>
<td>NOUN</td>
<td>DET</td>
<td></td>
</tr>
<tr>
<td>PROPN</td>
<td>NUM</td>
<td></td>
</tr>
<tr>
<td>VERB</td>
<td>PART</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRON</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCONJ</td>
<td>X</td>
</tr>
</tbody>
</table>
### Universal features

<table>
<thead>
<tr>
<th>Lexical features</th>
<th>Inflectional features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Nominal</strong></td>
</tr>
<tr>
<td>PronType</td>
<td>Gender</td>
</tr>
<tr>
<td>NumType</td>
<td>Animacy</td>
</tr>
<tr>
<td>Poss</td>
<td>Number</td>
</tr>
<tr>
<td>Reflex</td>
<td>Case</td>
</tr>
<tr>
<td>Foreign</td>
<td>Definite</td>
</tr>
<tr>
<td>Abbr</td>
<td>Degree</td>
</tr>
<tr>
<td></td>
<td>Polarity</td>
</tr>
<tr>
<td></td>
<td>Polite</td>
</tr>
</tbody>
</table>

**Index:** A abbreviation, abessive, ablative, absolute superlative, absolutive, accusative, active, additive, adessive, admirative, adverbal participle, affirmative, allative, animate, antipassive, aorist, article, aspect, associative, B benefactive, C cardinal, case, causative case, causative voice, collective noun, collective numeral, collective pronounal, comitative, common gender, comparative case, comparative degree, complex definiteness, conditional, conjunctive, construct state, converb, count plural, counting form, D dative, definite, definiteness, degree of comparison, delative, demonstrative, desiderative, destinative, direct case, direct voice, directional allative, distributive case, distributive numeral, dual, E elative, elevated referent, emphatic, equative
# Deprels (dependency relations)

<table>
<thead>
<tr>
<th></th>
<th>Nominals</th>
<th>Clauses</th>
<th>Modifier words</th>
<th>Function Words</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core arguments</strong></td>
<td>nsubj, obj, iobj</td>
<td>csubj, ccomp, xcomp</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Non-core dependents</strong></td>
<td>obl, vocative, expl, dislocated</td>
<td>advcl</td>
<td>advmod*, discourse</td>
<td>aux, cop, mark</td>
</tr>
<tr>
<td><strong>Nominal dependents</strong></td>
<td>nmod, appos, nummod</td>
<td>acl</td>
<td>amod</td>
<td>det, clf, case</td>
</tr>
<tr>
<td><strong>Coordination</strong></td>
<td>coni, cc</td>
<td>fixed, flat, compound</td>
<td>list, parataxis</td>
<td>orphan, goeswith, reparandum</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>punct, root, dep</td>
</tr>
</tbody>
</table>
UD v2 guidelines changes

- spaces in forms and lemmas (Vietnamese, “100 000”, “i.e.”)
- few tags, deprels and features renamed (CONJ→CCONJ, dobj→obj, Negative→Polarity, ...)
- new deprels (obl, clf), features (Polite, ...) and values
- removed deprels (neg, auxpass)
- copula verbs tagged as AUX
- coordinating conjunction & punctuation attached to the next conjunct (not the first conjunct)

UDv1:

UDv2:
UD v2 guidelines changes

- remnant-style ellipsis → orphan-style

UDv1:

UDv2:
empty nodes allowed in enhanced deps

basic:

enhanced:
UD v2 guidelines changes

- empty nodes allowed in enhanced deps
- DEPS column with all enhanced deps (not just the extra)

```
1  I   I  PRON  PRP  Number=Sing|Person=1|PronType=Prs  2  nsubj  2: nsubj _
2  like  like  VERB  VBP  Mood=Ind|Tense=Pres|VerbForm=Fin  0  root  0:root _
3  tea  tea  NOUN  NN  Number=Sing
4  and  and  CCONJ  CC  _
5  you  you  PRON  PRP  Case=Nom|Person=2|PronType=Prs  2  obj  2: obj _
5.1  _  _  VERB  VBP  Mood=Ind|Tense=Pres|VerbForm=Fin  _  _  2: conj _
6  rum  rum  NOUN  NN  Number=Sing
7  .  .  PUNCT  .  _
```
UD v2 guidelines changes

- 4 types of enhanced dependencies specified
  - controlled/raised subjects
  - ellipsis
  - propagation of conjunct
  - relative clauses
  - case information
UD v2 guidelines changes

- 4 types of enhanced dependencies specified
  - controlled/raised subjects
  - ellipsis
  - propagation of conjunct
  - relative clauses
  - case information

The diagram illustrates the relationships between the words in the sentences:

1. "The store buys and sells cameras.
2. "She was reading or watching a movie."
UD v2 guidelines changes

- 4 types of enhanced dependencies specified
  - controlled/raised subjects
  - ellipsis
  - propagation of conjunct
  - relative clauses
  - case information
4 types of enhanced dependencies specified
- controlled/raised subjects
- ellipsis
- propagation of conjunct
- relative clauses
- case information
UD v2 guidelines changes

- sent_id and text comments required (and SpaceAfter=No)

```plaintext
# sent_id = 1
# text = I like tea and you rum.
1 I I PRON PRP Number=Sing|Person=1|PronType=Prs 2 nsubj 2:nsubj _
2 like like VERB VBP Mood=Ind|Tense=Pres|VerbForm=Fin 0 root 0:root _
3 tea tea NOUN NN Number=Sing 2 obj 2:obj _
4 and and CCONJ CC _ 5 cc 5.1:cc _
5 you you PRON PRP Case=Nom|Person=2|PronType=Prs 2 conj 5.1:nsubj_ 5.1:obj _
5.1 _ _ VERB VBP Mood=Ind|Tense=Pres|VerbForm=Fin _ _ 2:conj _
6 rum rum NOUN NN Number=Sing 5 orphan 5.1:obj SpaceAfter=No
7 . . PUNCT . _ 2 punct 2:punct _
```
How simple is simple enough?
File format: CoNLL-U (1 file, 6 lines, 218 bytes)

<table>
<thead>
<tr>
<th>sent_id</th>
<th>text</th>
<th>1 John</th>
<th>John PROPN NNP Number=Sing</th>
<th>2 loves</th>
<th>love VERB VBZ Mood=Ind</th>
<th>Number=Sing</th>
<th>Person=3</th>
<th>Tense=Pres</th>
<th>VerbForm=Fin</th>
<th>3 Mary</th>
<th>Mary PROPN NNP Number=Sing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 John</td>
<td>John PROPN NNP Number=Sing</td>
<td>2 loves</td>
<td>love VERB VBZ Mood=Ind</td>
<td>Number=Sing</td>
<td>Person=3</td>
<td>Tense=Pres</td>
<td>VerbForm=Fin</td>
<td>3 Mary</td>
<td>Mary PROPN NNP Number=Sing</td>
</tr>
</tbody>
</table>
Prague Markup Language (PML)

Quotes from the newest documentation:

- PML is a common basis of an open family of XML-based data formats.
- PML is an on-going project in its early stage.

Facts:

- PML is used in Prague-* treebanks (and lexicons), Tamil, Indonesian, Latin and Ancient Greek DT, Lithuanian DT,...
- PML has just one implementation (Perl): Fslib/TrEd/Treex::PML
- PML is difficult to understand and maintain.
UD 2.0 & CoNLL-U vs. PDT 2.0 & PML

Guidelines (number of pages) – Let’s compare uncomparable...

<table>
<thead>
<tr>
<th></th>
<th>UD</th>
<th>PDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>3</td>
<td>53</td>
</tr>
<tr>
<td>morpho</td>
<td>61</td>
<td>56</td>
</tr>
<tr>
<td>syntax</td>
<td>60</td>
<td>317</td>
</tr>
<tr>
<td>tecto</td>
<td>–</td>
<td>1287</td>
</tr>
<tr>
<td>format</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>schema</td>
<td>–</td>
<td>59</td>
</tr>
<tr>
<td>total</td>
<td>131</td>
<td>1794</td>
</tr>
</tbody>
</table>


Guide “to become quickly familiar with the basic ideas”

“PML also tries to retain simplicity.”
UD 2.0 & CoNLL-U vs. PDT 2.0 & PML

Guidelines (number of pages) – Let’s compare incomparable...

<table>
<thead>
<tr>
<th></th>
<th>UD</th>
<th>PDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>3</td>
<td>53</td>
</tr>
<tr>
<td>morpho</td>
<td>61</td>
<td>56</td>
</tr>
<tr>
<td>syntax</td>
<td>60</td>
<td>317</td>
</tr>
<tr>
<td>tecto</td>
<td>–</td>
<td>1287</td>
</tr>
<tr>
<td>format</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>schema</td>
<td>–</td>
<td>59</td>
</tr>
<tr>
<td>total</td>
<td>131</td>
<td>1794</td>
</tr>
</tbody>
</table>

guide “to become quickly familiar with the basic ideas”

UD 2.0 & CoNLL-U vs. PDT 2.0 & PML

Guidelines (number of pages) – Let’s compare uncomparable...

<table>
<thead>
<tr>
<th></th>
<th>UD</th>
<th>PDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>3</td>
<td>53</td>
</tr>
<tr>
<td>morpho</td>
<td>61</td>
<td>56</td>
</tr>
<tr>
<td>syntax</td>
<td>60</td>
<td>317</td>
</tr>
<tr>
<td>tecto</td>
<td>–</td>
<td>1287</td>
</tr>
<tr>
<td>format</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>schema</td>
<td>–</td>
<td>59</td>
</tr>
<tr>
<td>total</td>
<td>131</td>
<td>1794</td>
</tr>
</tbody>
</table>

guide “to become quickly familiar with the basic ideas”

“PML also tries to retain simplicity.”

Simplicity?

- Simplicity is not only about file size or doc pages.
- Success is not only about simplicity.
- UD/CoNLL-U approach:
  
  Let simple things be ultra-simple
  (at the cost of edge cases being less elegant or not possible).
  - literal underscore token in CoNLL-U
  - original text including inter-sentence spaces
  - add new type of annotation
    (cross-sentence coreference, alignment)
Simplicity: Coordinations

PDT

- a-tree
  - zone=und
  - and
    - Coord
    - AuxK
  - store
    - Sb
    - Pred_Co
    - sells
      - Pred_Co
      - cameras
        - Obj
  - The
    - AuxA

UD basic

- a-tree
  - zone=und
  - buys
    - root
    - store
      - subj
      - sells
        - conj
        - cameras
          - obj
          - punct
  - The
    - det
    - and
    - cc
Simplicity: Coordinations

PDT

UD basic

UD enhanced
CoNLL 2017 multilingual parsing shared task

- http://universaldependencies.org/conll17/
- from Raw Text to Universal Dependencies
- segmentation, tokenization, labelled parsing (no morpho)
- 45 UDv2 languages plus X surprise languages
- registration deadline: April 15
- test phase: May 8–12
- organizers: ÚFAL, Google, Uppsala, Turku
- participants: 72 teams so far
Tools for annotating dependency trees

- **TrEd (+EasyTreex):** powerful, customizable, Perl, old
- **Brat:** online/JS+Python, UD support, embeddable
- **EasyTree:** perhaps too simple
- **GraphAnno:** useful for discourse etc., Java, keyboard
Tools for annotating dependency trees

- **TrEd (+EasyTreex)**: powerful, customizable, Perl, old
- **Brat**: online/JS+Python, UD support, embeddable
- **EasyTree**: perhaps too simple
- **GraphAnno**: useful for discourse etc., Java, keyboard
Tools for annotating dependency trees

- **TrEd (+EasyTreex):** powerful, customizable, Perl, old
- **Brat:** online/JS+Python, UD support, embeddable
- **EasyTree:** perhaps too simple
- **GraphAnno:** useful for discourse etc., Java, keyboard

![Dependency Tree Example](image-url)
Tools for annotating dependency trees

- **TrEd (+EasyTreex):** powerful, customizable, Perl, old
- **Brat:** online/JS+Python, UD support, embeddable
- **EasyTree:** perhaps too simple
- **GraphAnno:** useful for discourse etc., Java, keyboard
Automatic parsing: UDPipe

UDPipe by Milan Straka, try it online/as a webservice
http://lindat.mff.cuni.cz/services/udpipe/

- End-to-end, batteries included:
  segment, tokenize, tag, morpho, lemma, labelled parsing
- Pretrained models for all the UD (2.0 soon) langs
- User friendly (outputs CoNLL-U, Table, SVG)
- State-of-the-art quality, ultra fast
- Open-source, easy install for Linux, OS X, Win
- Interfaces for C++, C#, Java, Perl, Python
- Easily train on your own data
Tools for viewing dependency trees

- all the editors, including UDPipe online
- PML-TQ (UDv1.2)
  https://lindat.mff.cuni.cz/services/pmltq/
- Udapi https://github.com/udapi/udapi-python
  udapy Write::HTML < my.conllu > my.html
Tools for viewing dependency trees


```
zone=cs
id=f000001-s1/cs

jsem
VERB
root

Kde
ADV
advmod

PUNCT
punct

 zone=en
id=f000001-s1/en

am
VERB
root

Where
ADV
dep

I
PRON
nsubj

PUNCT
punct

 lemma=be
Tense=Pres
VerbForm=Fin
```
Tools for viewing dependency trees

- all the editors, including UDPipe online
- PML-TQ (UDv1.2)
  https://lindat.mff.cuni.cz/services/pmltq/
- Udapi https://github.com/udapi/udapi-python

udapy Write::HTML < my.conllu > my.html
demo: http://ufallab.ms.mff.cuni.cz/~popel/czeng1.6-sample.html

udapy -HA < my.conllu > my.html
Tools for viewing dependency trees

bugs = ud.MarkBugs

Det den DET DT|NEU|SIN|DEF Definite=Def|Gender=Neut|Number=Sing|PronType=Prs det

svedvridna snedvrida ADJ PC|PRF|UTR/NEU|SIN|DEF|NOM Case=Nom|Definite=Def|Number=Sing|Tense=P

livsperspektiv livsperspektiv NOUN NN|NEU|SIN|IND|NOM Case=Nom|Definite=Ind|Gender=Neut|Number=1

som som PRON HP|Rel|obj

många många ADJ JJ|POS|UTR/NEU|PLU|IND/DEF|NOM Case=Nom|Degree=Pos|Number=Plur amod _
män man NOUN NN|UTR|PLU|IND|NOM Case=Nom|Definite=Ind|Gender=Com|Number=Plur nsubj _

får få VERB VB|PRS|AKT Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act acl:relcl _
i i ADP PP _ advmod

dag dag NOUN NN|UTR|SIN|IND|NOM Case=Nom|Definite=Ind|Gender=Com|Number=Sing fixed _
genom genom ADP PP _ mark _

att att PART IE _ mark _

aldrig aldrig ADV AB Polarity=Neg advmod

befatta befatta VERB VB|INF|AKT VerbForm=Inf|Voice=Act advcl _
sig sig PRON PN|UTR/NEU|SIN/PLU|DEF|OBJ Case=Acc|Definite=Def|PronType=Prs obj _

med med ADP PP _ case _

' ' PUNCT PAD _ punct SpaceAfter=No

de de en DET DT|UTR/NEU|PLU|DEF Definite=Def|Number=Plur|PronType=Art det _
bakre bakre ADJ JJ|KOM|UTR/NEU|SIN|IND|PLU|IND/DEF|NOM Case=Nom|Degree=Cmp amod

linjerna line NOUN NN|UTR|PLU|DEF|NOM Case=Nom|Definite=Def|Gender=Com|Number=Plur obl S

' ' PUNCT PAD _ punct

( ( PUNCT PAD _ punct SpaceAfter=No

!! PUNCT PAD _ punct Bug=punct-child|SpaceAfter=No

)) PUNCT PAD _ punct Bug=punct-child

änder var AUX VB|PRS|AKT Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act cop _

förvisso förvisso ADV AB _ advmod _
Tools for viewing dependency trees

- all the editors, including UDPipe online
- PML-TQ (UDv1.2)
  https://lindat.mff.cuni.cz/services/pmltq/
- Udapi https://github.com/udapi/udapi-python

udapy Write::HTML < my.conllu > my.html
demo: http://ufallab.ms.mff.cuni.cz/~popel/czeng1.6-sample.html

udapy -HA < my.conllu > my.html
udapy -T < my.conllu | less -R
Tools for viewing dependency trees

```
# sent_id = 1
# text = Corriere Sport da pagina 23 a pagina 26

Corriere PROPN root
  Sport PROPN name
    da ADP case
  pagina NOUN nmod
    23 NUM nummod
    a ADP case
  pagina NOUN nmod
    26 NUM nummod

# sent_id = 2
# text = I tre avevano da poco lasciato la cima e stavano cominciando la discesa.

I DET det
  tre NUM nsubj
  avevano AUX aux
    da ADP case
    poco ADV advmod
  lasciato VERB root
    la DET det
    cima NOUN dobj
    e CONJ cc
    stavano AUX aux
    cominciando VERB conj
      la DET det
      discesa NOUN dobj
    . PUNCT punct
```
Tools for viewing dependency trees

- all the editors, including UDPipe online
- PML-TQ (UDv1.2)
  https://lindat.mff.cuni.cz/services/pmltq/
- Udapi https://github.com/udapi/udapi-python

udapy Write::HTML < my.conllu > my.html

demo: http://ufallab.ms.mff.cuni.cz/~popel/czeng1.6-sample.html

udapy -HA < my.conllu > my.html

udapy -T < my.conllu | less -R
T=text, H=html, A=all attributes, N=no color, M=marked
Udapi

- [link](http://udapi.github.io/)
- API and multi-language framework for processing UD
- Allows both fast prototyping and full applications
- Both command-line tool (`udapy`) and library
- Modularity, reusability, cooperation
Udapi use cases

- format conversions (CoNLL-U, SDParse, PML, VISL-cg, TikZ)
- ud.Convert1to2 transformations from UD v1 to v2
  used for Bulgarian, Romanian, Galician, Russian, Irish, ...
- ud.MarkBugs validity tests
- ud.SetSpaceAfter, ud.SetSpaceAfterFromText
- util.Eval, util.Filter, util.Wc
- automatic parsing (via UDPipe), evaluation,...

Hands-on tutorial

http://udapi.github.io/tutorial/
Treex vs. Udapi

Treex (2005)
- Perl
- over 10 ÚFAL developers
- multilingual in the end
- Prague-style dependencies
- tectogrammatical layer
- PML / XML / *.treex
- weak UD support
- focus: TectoMT
  
  https://github.com/ufal/treex

Udapi (2016)
- Python, Perl, Java
- MP+ZŽ+M.Vojtek
- multilingual from the start
- Universal dependencies
- no layers (but zones)
- CoNLL-U
- MWT, empty, enhanced
- focus: speed & simplicity

  https://github.com/udapi/
Benchmark: Speed-up relative to Treex

(source: https://github.com/martinpopel/newtreex)
Benchmark: Memory (MB)

(source: https://github.com/martinpopel/newtreex)
cs-ud-train-1.conllu: 68 MiB, 41k sentences, 0.8 MWords
Algorithmic challenges

- data structure for globally-ordered rooted trees
  node.descendants ...ordered
  node.shift_before_node()

- efficient loading&saving of CoNLL-U files
  linear-time checking of cycles
  lazy deserialization of FEATS and MISC

- write.TextModeTrees for non-projective trees
  minimize crossings and/or depth

- align raw sentence with gold-annotated words
  add SpaceAfter=No and goeswith
  create multi-word tokens (minimal)
  mark typos etc. in MISC
Summary

EVERY NOW AND THEN I REALIZE I'M MAINTAINING A HUGE CHAIN OF TECHNOLOGY SOLELY TO SUPPORT ITSELF.

https://www.xkcd.com/1579/